Browsable Borland Delphi Assembler Grammar

Grammar connected by Vadim Zaytsev, see the Grammar Zoo entry for details: assembly/delphi/txl/cangas/connected
Source used for this grammar: Jorge L. Cangas, *TXL Grammar for Borland Delphi 2006,* `asm.grammar`, December 2007

Summary

- Total 29 production rules with 238 top alternatives and 402 symbols.
- Vocabulary: 232 = 37 nonterminals + 195 terminals + 0 labels + 0 markers.
- Total 37 nonterminal symbols: 29 defined (asm_stmtblock, asm_stm, asmlabel_colon, asm_unlabeledstm_semi, asmid, asmlabel, asm_expr, asm_infix_expr, asm_term, asm_primary, asmhex, asm_unlabeledstm, asm_opcode_prefix, asm_opcode, asm_directive, asm_directive_arg, asm_register, segmt_overr, asm_prefixop, asm_infixop, asm_postfixop, asmlbl, SP, NL, end_struct, sign, colon, label_id, anynumber), 1 root (asm_stmtblock), 0 top (—), 8 bottom (key, id [4], integernumber, space, stringlit, number [2], hexnumber, charlit [2]).
- Total 195 terminal symbols: 170 keywords (*"asm"*, *"lock"*, *"adc"*, *"add"*, *"and"* [2], *"bswap"*, *"bt"*, *"btr"*, *"bts"*, *"call"*, *"cdq"*, *"cld"*, *"cmp"*, *"dec"*, *"div"*, *"fabs"*, *"fadd"*, *"faddp"*, *"fbstp"*, *"fchs"*, *"fclex"*, *"fcom"*, *"fcomp"*, *"fcos"*, *"fdiv"*, *"fdivp"*, *"fdivrp"*, *"ffree"*, *"fiadd"*, *"fidiv"*, *"fild"*, *"fimul"*, *"fistp"*, *"fld"*, *"fldcw"*, *"fldz"*, *"fmul"*, *"fmulp"*, *"fnclex"*, *"fninit"*, *"fnstcw"*, *"fnstsw"*, *"fpatan"*, *"fprem"*, *"fptan"*, *"frndint"*, *"fscale"*, *"fsin"*, *"fsincos"*, *"fsqrt"*, *"fstcw"*, *"fstp"*, *"fstsw"*, *"fsub"*, *"fsubp"*, *"fsubr"*, *"fwait"*, *"fxch"*, *"fxtract"*, *"imul"*, *"inc"*, *"int"*, *"ja"*, *"jae"*, *"jb"*, *"jbe"*, *"jc"*, *"je"*, *"jecxz"*, *"jg"*, *"jge"*, *"jl"*, *"jle"*, *"jmp"*, *"jnc"*, *"jne"*, *"jnl"*, *"jns"*, *"jnz"*, *"jo"*, *"jp"*, *"js"*, *"jz"*, *"lea"*, *"leave"*, *"lodsb"*, *"lodsw"*, *"loop"*, *"mov"*, *"movsb"*, *"movsx"*, *"movzx"*, *"mul"*, *"neg"*, *"not"* [2], *"or"* [2], *"pop"*, *"popfd"*, *"push"*, *"pushfd"*, *"rcl"*, *"rcr"*, *"rep"*, *"repe"*, *"repne"*, *"ret"*, *"rol"*, *"ror"*, *"sahf"*, *"sar"*, *"sbb"*, *"seto"*, *"shl"* [2], *"shld"*, *"shr"* [2], *"shrd"*, *"std"*, *"stosb"*, *"stosd"*, *"stosw"*, *"sub"*, *"test"*, *"wait"*, *"xadd"*, *"xchg"*, *"xor"* [2], *"DB"*, *"DW"*, *"DD"*, *"DQ"*, *"ST"* [2], *"FS"* [2], *"GS"* [2], *"EAX"*, *"EBX"*, *"ECX"*, *"EDX"*, *"ESP"*, *"EBP"*, *"ESI"*, *"EDI"*, *"AX"*, *"BX"*, *"CX"*, *"DX"*, *"SP"*, *"BP"*, *"SI"*, *"DI"*, *"AL"*, *"BL"*, *"CL"*, *"DL"*, *"CS"* [2], *"DS"* [2], *"SS"* [2], *"ES"* [2], *"AH"*, *"BH"*, *"CH"*, *"DH"*, *"high"*, *"low"*, *"offset"*, *"dmtindex"*, *"vmtoffset"*, *"type"*, *"ptr"*, *"mod"*, *"end"*), 0 letters (—), 0 numerics (—), 16 signs (*";"*, *"@"* [2], *"@+"* [3], *"["* [2], *"]"* [2], *"("* [2], *")"* [2], *","* [2], *":"* [2], *"&"*, *"."* [2], *"+"* [2], *"-"* [2], *"*"*, *"/"*, *" "*).

Syntax

```
asm_stmtblock ::=
        "asm" asm_stm* end_struct
```

```
asm_stm ::=
        asmlabel_colon? asm_unlabeledstm_semi?
```

```
asmlabel_colon ::=
        asmlabel colon
```

```
asm_unlabeledstm_semi ::=
        asm_unlabeledstm ";"? NL
```

```
asmid ::=
        "@"* asmlbl
        "@+"* id
        "@+"* anynumber
        "@+"* key
```

```
asmlabel ::=
        asmid+
        label_id
```

```
asm_expr ::=
        asm_term asm_infix_expr*
```

```
asm_infix_expr ::=
        asm_infixop asm_term
```

```
asm_term ::=
        asm_prefixop* asm_primary asm_postfixop*
```

```
asm_primary ::=
        "[" asm_expr "]"
        "(" asm_expr ")"
        asm_register
        id
        anynumber
        charlit
        stringlit
        asmhex
        SP asmlabel
```

```
asmhex ::=
        number id
```

```
asm_unlabeledstm ::=
        asm_directive {asm_directive_arg ","}*
        asm_opcode_prefix? asm_opcode {asm_expr ","}*
```

```
asm_opcode_prefix ::=
        "lock" space?
```

```
asm_opcode ::=
        "adc"
        "add"
        "and"
        "bswap"
        "bt"
        "btr"
        "bts"
        "call"
        "cdq"
        "cld"
        "cmp"
        "dec"
        "div"
        "f2xm1"
        "fabs"
        "fadd"
        "faddp"
        "fbstp"
        "fchs"
        "fclex"
        "fcom"
        "fcomp"
        "fcos"
        "fdiv"
        "fdivp"
        "fdivrp"
        "ffree"
        "fiadd"
        "fidiv"
        "fild"
        "fimul"
        "fistp"
        "fld"
        "fld1"
        "fldcw"
        "fldl2e"
        "fldlg2"
        "fldln2"
        "fldz"
        "fmul"
        "fmulp"
        "fnclex"
        "fninit"
        "fnstcw"
```

```
"fnstsw"
"fpatan"
"fprem"
"fptan"
"frndint"
"fscale"
"fsin"
"fsincos"
"fsqrt"
"fstcw"
"fstp"
"fstsw"
"fsub"
"fsubp"
"fsubr"
"fwait"
"fxch"
"fxtract"
"fyl2x"
"fyl2xp1"
"imul"
"inc"
"int"
"ja"
"jae"
"jb"
"jbe"
"jc"
"je"
"jecxz"
"jg"
"jge"
"jl"
"jle"
"jmp"
"jnc"
"jne"
"jnl"
"jns"
"jnz"
"jo"
"jp"
"js"
"jz"
"lea"
```

```
        "leave"
        "lodsb"
        "lodsw"
        "loop"
        "mov"
        "movsb"
        "movsx"
        "movzx"
        "mul"
        "neg"
        "not"
        "or"
        "pop"
        "popfd"
        "push"
        "pushfd"
        "rcl"
        "rcr"
        "rep"
        "repe"
        "repne"
        "ret"
        "rol"
        "ror"
        "sahf"
        "sar"
        "sbb"
        "seto"
        "shl"
        "shld"
        "shr"
        "shrd"
        "std"
        "stosb"
        "stosd"
        "stosw"
        "sub"
        "test"
        "wait"
        "xadd"
        "xchg"
        "xor"
```

```
asm_directive ::=
        "DB"
```

```
        "DW"
        "DD"
        "DQ"
```

```
asm_directive_arg ::=
        charlit
        sign? anynumber
        asm_expr
```

```
asm_register ::=
        "ST" "(" integernumber ")"
        "ST"
        "FS"
        "GS"
        "EAX"
        "EBX"
        "ECX"
        "EDX"
        "ESP"
        "EBP"
        "ESI"
        "EDI"
        "AX"
        "BX"
        "CX"
        "DX"
        "SP"
        "BP"
        "SI"
        "DI"
        "AL"
        "BL"
        "CL"
        "DL"
        "CS"
        "DS"
        "SS"
        "ES"
        "AH"
        "BH"
        "CH"
        "DH"
        "CS" segmt_overr?
        "DS" segmt_overr?
        "SS" segmt_overr?
```

```
        "FS" segmt_overr?
        "GS" segmt_overr?
        "ES" segmt_overr?
```

```
segmt_overr ::=
        ":" asm_expr
```

```
asm_prefixop ::=
        "high"
        "low"
        "offset"
        "dmtindex"
        "vmtoffset"
        "type"
        "not"
        "&"
        sign
        "@"
```

```
asm_infixop ::=
        "."
        "+"
        "-"
        "*"
        "/"
        "ptr"
        "mod"
        "xor"
        "and"
        "or"
        "shr"
        "shl"
```

```
asm_postfixop ::=
        "[" asm_expr "]"
        "." asm_expr
```

```
asmlbl ::=
        "@(\\d+\\u+\\i*)"
```

```
SP ::=
        " "
```

```
NL ::=
```

```
        "\\n"
```

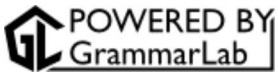
```
end_struct ::=
        "end"
```

```
sign ::=
        "+"
        "-"
```

```
colon ::=
        ":"
```

```
label_id ::=
        anynumber
        id
```

```
anynumber ::=
        hexnumber
        number
```



Maintained by Dr. Vadim Zaytsev a.k.a. @grammarware. Last updated in September 2015. [↑]